

# The Hidden Docs In Angular



/ 谢亚东 (执衡)  
Senior Front-end Engineer  
Alibaba





谢亚东 (执衡)  
Senior Front-end Engineer  
Alibaba



# 谢亚东 (执衡)

## Senior Front-end Engineer

### Alibaba

## 实时计算

实时计算 (Alibaba Cloud Realtime Compute, Powered by Ververica) 是阿里云提供的基于 Apache Flink 构建的企业级大数据计算平台。在 PB 级别的数据集上可以支持亚秒级别的处理延时，赋能用户标准实时数据处理流程和行业解决方案；支持 Datastream API 作业开发，提供了批流统一的 Flink SQL，简化 BI 场景下的开发；可与用户已使用的大数据组件无缝对接，更多增值特性助力企业实时化转型。

The screenshot shows the Alibaba Cloud Realtime Compute (Flink) product page. At the top, there are navigation tabs: '独享模式' (Exclusive Mode), 'Flink云原生' (Flink Native Cloud), '共享模式' (Shared Mode), '案例与解决方案' (Case Studies and Solutions), '产品文档' (Product Documentation), '实时数据分析及大屏展示' (Real-time Data Analysis and Big Screen Display), and 'Flink Forward Asia大会' (Flink Forward Asia Conference). Below the navigation, there's a diagram illustrating the architecture with components like 'Data Lake功能' (Data Lake Function), 'SQL化' (SQLization), 'UDX开放' (UDX Open), and '开发平台' (Development Platform). A large orange box highlights 'SQL化' and 'Data Lake功能'. The background features a dark theme with abstract data visualization elements.

实时计算 (Alibaba Cloud Realtime Compute, Powered by Ververica) 是阿里云提供的基于 Apache Flink 构建的企业级大数据计算平台。在 PB 级别的数据集上可以支持亚秒级别的处理延时，赋能用户标准实时数据处理流程和行业解决方案；支持 Datastream API 作业开发，提供了批流统一的 Flink SQL，简化 BI 场景下的开发；可与用户已使用的大数据组件无缝对接，更多增值特性助力企业实时化转型。

独享模式 Flink云原生 共享模式 案例与解决方案 产品文档 实时数据分析及大屏展示 Flink Forward Asia大会

Data Lake功能

SQL化

UDX开放

开发平台

Data Lake功能

独享模式支持DataLake场景下的数据清洗，数据分析，数据同步，异构数据源计算等功能

开发平台

UDX开放

独享模式可开放UDF/UDAF/UDTF接口，为用户提供更强大的数据处理能力。



# 谢亚东 (执衡)

## Senior Front-end Engineer

### Alibaba

实时计算

实时计算 (Alibaba Cloud Realtime Computing)  
Flink 构建的企业级大数据计算平台  
准实时数据处理流程和行业解决方案  
简化 BI 场景下的开发；可与用户直连

独享模式 Flink

Data Lake

UDX开放

开发平台



What is Apache Flink?

Use Cases

Powered By

FAQ

Downloads

Tutorials ↗

Documentation ▾

Getting Help

Flink Blog

Ecosystem

Community & Project Info

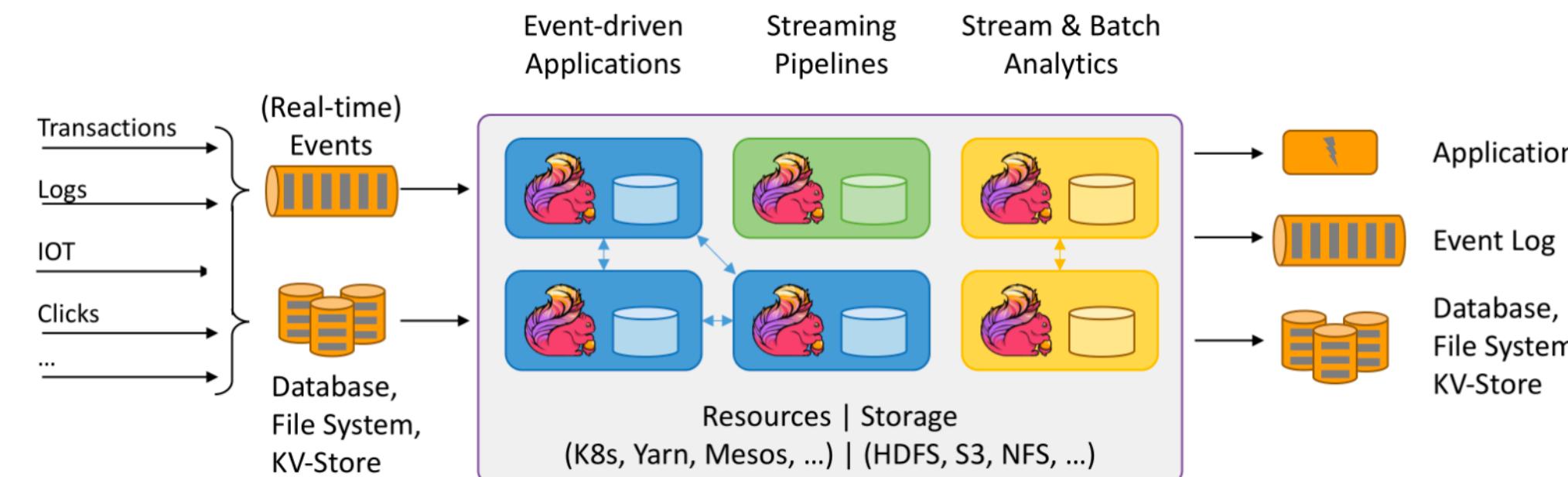
Roadmap

How to Contribute

Flink on GitHub ↗

中文版

## Apache Flink® – Stateful Computations over Data Streams



### All streaming use cases

- Event-driven Applications
- Stream & Batch Analytics
- Data Pipelines & ETL

[Learn more](#)

### Guaranteed correctness

- Exactly-once state consistency
- Event-time processing
- Sophisticated late data handling

[Learn more](#)

### Layered APIs

- SQL on Stream & Batch Data
- DataStream API & DataSet API
- ProcessFunction (Time & State)

[Learn more](#)

### Operational Focus

- Flexible deployment
- High-availability setup
- Savepoints

[Learn more](#)

### Scales to any use case

- Scale-out architecture
- Support for very large state
- Incremental checkpointing

[Learn more](#)

### Excellent Performance

- Low latency
- High throughput
- In-Memory computing

[Learn more](#)



# 谢亚东 (执衡)

## Senior Front-end Engineer

### Alibaba

实时计算

实时计算 (Alibaba Cloud Realtime Computing)  
Flink 构建的企业级大数据计算平台  
准实时数据处理流程和行业解决方案  
简化 BI 场景下的开发；可与用户直连

独享模式

Flink

Data Lake

UDX开放

开发平台



What is Apache Flink?

Use Cases

Powered By

FAQ

Downloads

Tutorials ↗

Documentation ▾

Getting Help

Flink Blog

Ecosystem

Community & Project Info

Roadmap

How to Contribute

Flink on GitHub ↗

中文版



Quick Start

Download ▾

Docs ▾

Helium

Community ▾

Apache ▾

## Apache Zeppelin

Web-based notebook that enables data-driven,  
interactive data analytics and collaborative documents with SQL, Scala and more.

GET STARTED

DOWNLOAD

## TECHNOLOGIES



SQL



Code

Issues 218

Pull requests 31

Actions

Projects 2

Wiki

Security

Insights

Settings

An enterprise-class UI components based on Ant Design and Angular. 🐍 <https://ng.ant.design>

1,224 commits

1 branch

0 packages

81 releases

2 environments

105 contributors

MIT



# NG-ZORRO

An enterprise-class UI components based on Ant Design and Angular.

codefactor

A

build

passing

coverage

92%

release date

last monday

npm

v8.5.1

downloads

87k/month

license

MIT

chat

on gitter

extension for [VSCode](#) [Twitter](#) [NG-ZORRO](#)

Success Tips

Informational Notes

Warning

Error

Primary

Normal

Dashed

Danger

Download

&lt; Backward

Forward &gt;

\* Select: Please select a country

Select[multiple]: Please select favourite colors

InputNumber: 3 machines

magenta red volcano orange gold lime

green cyan blue geekblue purple

Finished  
In Progress  
This is a description. This is a description.Switch:

# Angular Evolution

2.0    4.0    5.0    6.0    7.0    8.0    9.0

2016 Sep

2019 Nov

# Code After Compiler

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  template: `<div><h2>Hello {{name}}</h2></div>`,
})
export class AppComponent {
  name = 'Angular';
}
```

# Code After Compiler

2.0

```
(function anonymous(jit_StaticNodeDebugInfo0,jit_createRenderComponentType1,
  jit_22,jit_DebugAppView3,jit_14,jit_CD_INIT_VALUES,
  jit_createRenderElement6,jit__object_Object_7,jit_inlineInterpolate8,jit_checkBinding9
) {
  var styles_App = [];
  var nodeDebugInfos_App0 = [
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{})
  ]
;
  var renderType_App = jit_createRenderComponentType1('./App class App - inline template',0,jit_22,styles_App,{});
  function View_App0(viewUtils,parentView,parentIndex,parentElement) {
    var self = this;
    jit_DebugAppView3.call(this, View_App0,renderType_App,jit_14,viewUtils,
      parentView,parentIndex,parentElement,jit_22,nodeDebugInfos_App0);
    self._expr_7 = jit_CD_INIT_VALUES;
  }
  View_App0.prototype = Object.create(jit_DebugAppView3.prototype);
  View_App0.prototype.createInternal = function(rootSelector) {
    var self = this;
    var parentRenderNode = self.renderer.createViewRoot(self.parentElement);
    self._text_0 = self.renderer.createText(parentRenderNode,'\\n  ',self.debug(0,0,0));
    self._el_1 = jit_createRenderElement6(self.renderer,parentRenderNode,'div',jit__object_Object_7,self.debug(1,1,4));
    self._text_2 = self.renderer.createText(self._el_1,'\\n  ',self.debug(2,1,9));
    self._el_3 = jit_createRenderElement6(self.renderer,self._el_1,'h2',jit__object_Object_7,self.debug(3,2,6));
    self._text_4 = self.renderer.createText(self._el_3,'',self.debug(4,2,10));
    self._text_5 = self.renderer.createText(self._el_1,'\\n  ',self.debug(5,2,29));
    self._text_6 = self.renderer.createText(parentRenderNode,'\\n  ',self.debug(6,3,10));
    self.init(null,(self.renderer.directRenderer? null:[
      self._text_0,
      self._el_1,
      self._text_2,
      self._el_3,
      self._text_4,
      self._text_5,
      self._text_6
    ]),
    null);
    return null;
  };
  View_App0.prototype.detectChangesInternal = function(throwOnChange) {
    var self = this;
    self.debug(4,2,10);
    var currVal_7 = jit_inlineInterpolate8(1,'Hello ',self.context.name,'');
    if (jit_checkBinding9(throwOnChange,self._expr_7,currVal_7)) {
      self.renderer.setText(self._text_4,currVal_7);
      self._expr_7 = currVal_7;
    }
  };
  return View_App0
})
```

# Code After Compiler

2.0

```
(function anonymous(jit_StaticNodeDebugInfo0,jit_createRenderComponentType1,
  jit_22,jit_DebugAppView3,jit_14,jit_CD_INIT_VALUES,
  jit_createRenderElement6,jit__object_Object_7,jit_inlineInterpolate8,jit_checkBinding9
) {
  var styles_App = [];
  var nodeDebugInfos_App0 = [
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{})
  ]
;
  var renderType_App = jit_createRenderComponentType1('./App class App - inline t
function View_App0(viewUtils,parentView,parentIndex,parentElement) {
  var self = this;
  jit_DebugAppView3.call(this, View_App0,renderType_App,jit_14,viewUtils,
    parentView,parentIndex,parentElement,jit_22,nodeDebugInfos_App0);
  self._expr_7 = jit_CD_INIT_VALUES;
}
View_App0.prototype = Object.create(jit_DebugAppView3.prototype);
View_App0.prototype.createInternal = function(rootSelector) {
  var self = this;
  var parentRenderNode = self.renderer.createViewRoot(self.parentElement);
  self._text_0 = self.renderer.createText(parentRenderNode,'
',self.debug(4,2,10));
  self._el_1 = jit_createRenderElement6(self.renderer,parentRenderNode,'h2');
  self._text_2 = self.renderer.createText(self._el_1,'
',self.debug(4,2,10));
  self._el_3 = jit_createRenderElement6(self.renderer,self._el_1,'h2');
  self._text_4 = self.renderer.createText(self._el_3,'',self.debug(4,2,10));
  self._text_5 = self.renderer.createText(self._el_1,'
',self.debug(5,2,29));
  self._text_6 = self.renderer.createText(parentRenderNode,'
',self.debug(6,2,29));
  self.init(null,(self.renderer.directRenderer? null:[
    self._text_0,
    self._el_1,
    self._text_2,
    self._el_3,
    self._text_4,
    self._text_5,
    self._text_6
  ]),
  null);
  return null;
};
View_App0.prototype.detectChangesInternal = function(throwOnChange) {
  var self = this;
  self.debug(4,2,10);
  var currVal_7 = jit_inlineInterpolate8(1,'Hello ',self.context.name,'');
  if (jit_checkBinding9(throwOnChange,self._expr_7,currVal_7)) {
    self.renderer.setText(self._text_4,currVal_7);
    self._expr_7 = currVal_7;
  }
};
return View_App0
})
```

4.0

```
(function anonymous(jit_createRendererType2_0,jit_viewDef_1,jit_textDef_2,jit_elementDef_3) {
  var styles_App = [];
  var RenderType_App = jit_createRendererType2_0({encapsulation:2,styles:styles_App,
    data:{}});
  function View_App_0(_l) {
    return jit_viewDef_1(0,[
      (_l()),jit_textDef_2(-1,null,['\n  ']),
      (_l()),jit_elementDef_3(1,0,null,null,4,'div',[],null,null,null,null),
      (_l()),jit_textDef_2(-1,null,['\n  ']),
      (_l()),jit_elementDef_3(3,0,null,null,1,'h2',[],null,null,null,null),
      (_l()),jit_textDef_2(4,null,['Hello ','']),
      (_l()),jit_textDef_2(-1,null,['\n  ']),
      (_l()),jit_textDef_2(-1,null,['\n  '])
    ],
    null,
    function(_ck, _v) {
      var _co = _v.component;
      var currVal_0 = _co.name;
      _ck(_v,4,0,currVal_0);
    });
  }
  return {RenderType_App:RenderType_App,View_App_0:View_App_0};
})
```

# Code After Compiler

2.0

```
(function anonymous(jit_StaticNodeDebugInfo0,jit_createRenderComponentType1,
  jit_22,jit_DebugAppView3,jit_14,jit_CD_INIT_VALUES,
  jit_createRenderElement6,jit_object_Object_7,jit_inlineInterpolate8,jit_checkBinding9
) {
  var styles_App = [];
  var nodeDebugInfos_App0 = [
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{})
  ]
;
  var renderType_App = jit_createRenderComponentType1('./App class App - inline t
function View_App0(viewUtils,parentView,parentIndex,parentElement) {
  var self = this;
  jit_DebugAppView3.call(this, View_App0,renderType_App,jit_14,viewUtils,
    parentView,parentIndex,parentElement,jit_22,nodeDebugInfos_App0);
  self._expr_7 = jit_CD_INIT_VALUES;
}
View_App0.prototype = Object.create(jit_DebugAppView3.prototype);
View_App0.prototype.createInternal = function(rootSelector) {
  var self = this;
  var parentRenderNode = self.renderer.createViewRoot(self.parentElement);
  self._text_0 = self.renderer.createText(parentRenderNode,'
  ',self.debug(4,2,10));
  self._el_1 = jit_createRenderElement6(self.renderer,parentRenderNode,'
');
  self._text_2 = self.renderer.createText(self._el_1,'
  ',self.debug(4,2,10));
  self._el_3 = jit_createRenderElement6(self.renderer,self._el_1,'h2','');
  self._text_4 = self.renderer.createText(self._el_3,'',self.debug(4,2,10));
  self._text_5 = self.renderer.createText(self._el_1,'
  ',self.debug(5,2,29));
  self._text_6 = self.renderer.createText(parentRenderNode,'
  ',self.debug(6,2,29));
  self.init(null,(self.renderer.directRenderer? null:[
    self._text_0,
    self._el_1,
    self._text_2,
    self._el_3,
    self._text_4,
    self._text_5,
    self._text_6
  ]),
  null);
  return null;
};
View_App0.prototype.detectChangesInternal = function(throwOnChange) {
  var self = this;
  self.debug(4,2,10);
  var currVal_7 = jit_inlineInterpolate8(1,'Hello ',self.context.name,'');
  if (jit_checkBinding9(throwOnChange,self._expr_7,currVal_7)) {
    self.renderer.setText(self._text_4,currVal_7);
    self._expr_7 = currVal_7;
  }
};
return View_App0
})
```

4.0

```
(function anonymous(jit_createRendererType2_0,jit_viewDef_1,jit_textDef_2,jit_elementDef_3) {
  var styles_App = [];
  var RenderType_App = jit_createRendererType2_0({encapsulation:2,styles:styles_App,
    data:{}});
  function View_App_0(_1) {
    return jit_viewDef_1(0,[
      (_1()),jit_textDef_2(-1,null,['\n  ']),
      (_1()),jit_elementDef_3(1,0,null,null,4,'div',[],null,null,null,null),
      (_1()),jit_textDef_2(-1,null,['\n  ']),
      (_1()),jit_elementDef_3(3,0,null,null,1,'h2',[],null,null,null,null),
      (_1()),jit_textDef_2(4,null,['Hello ''']),
      (_1()),jit_textDef_2(-1,null,['\n  ']),
      (_1()),jit_textDef_2(-1,null,['\n  '])
    ],
    null,
    function(_ck, _v) {
      var _co = _v.component;
      var currVal_0 = _co.name;
      _ck(_v,4,0,currVal_0);
    });
    return {RenderType_App:RenderType_App,View_App_0:View_App_0};
  }
})
```

6.0

```
(function anonymous(i0) {
  var App = (function () {
    function App() {}
    App.ngComponentDef = i0.eDefineComponent({
      type: App,
      selectors: [['my-app']],
      factory: function App_Factory() {
        return new App();
      },
      template: function App_Template(ctx, cm) {
        if (cm) {
          i0.eE(0, 'div');
          i0.eE(1, 'h2');
          i0.eT(2);
          i0.ee();
          i0.ee();
        }
        i0.et(2, i0.ei1('Hello ', ctx.name, ''));
      }
    });
    return App;
  }());
  i0.erenderComponent(App);
})
```

# Code After Compiler

2.0

```
(function anonymous(jit_StaticNodeDebugInfo0,jit_createRenderComponentType1,
  jit_22,jit_DebugAppView3,jit_14,jit_CD_INIT_VALUES,
  jit_createRenderElement6,jit__object_Object_7,jit_inlineInterpolate8,jit_checkBinding9
) {
  var styles_App = [];
  var nodeDebugInfos_App0 = [
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{}),
    new jit_StaticNodeDebugInfo0([],null,{})
  ]
;
var renderType_App = jit_createRenderComponentType1('./App class App - inline t
function View_App0(viewUtils,parentView,parentIndex,parentElement) {
  var self = this;
  jit_DebugAppView3.call(this, View_App0,renderType_App,jit_14,viewUtils,
    parentView,parentIndex,parentElement,jit_22,nodeDebugInfos_App0);
  self._expr_7 = jit_CD_INIT_VALUES;
}
View_App0.prototype = Object.create(jit_DebugAppView3.prototype);
View_App0.prototype.createInternal = function(rootSelector) {
  var self = this;
  var parentRenderNode = self.renderer.createViewRoot(self.parentElement);
  self._text_0 = self.renderer.createText(parentRenderNode,'\\n  ',self.debug(4,2,10));
  self._el_1 = jit_createRenderElement6(self.renderer,parentRenderNode,'h2');
  self._text_2 = self.renderer.createText(self._el_1,'\\n  ',self.debug(4,2,10));
  self._el_3 = jit_createRenderElement6(self.renderer,self._el_1,'h2',j
  self._text_4 = self.renderer.createText(self._el_3,'',self.debug(4,2,10));
  self._text_5 = self.renderer.createText(self._el_1,'\\n  ',self.debug(5,2,29));
  self._text_6 = self.renderer.createText(parentRenderNode,'\\n  ',self.debug(6,
  self.init(null,(self.renderer.directRenderer? null: [
    self._text_0,
    self._el_1,
    self._text_2,
    self._el_3,
    self._text_4,
    self._text_5,
    self._text_6
  ]),
  null);
  return null;
};
View_App0.prototype.detectChangesInternal = function(throwOnChange) {
  var self = this;
  self.debug(4,2,10);
  var currVal_7 = jit_inlineInterpolate8(1,'Hello ',self.context.name,'');
  if (jit_checkBinding9(throwOnChange,self._expr_7,currVal_7)) {
    self.renderer.setText(self._text_4,currVal_7);
    self._expr_7 = currVal_7;
  }
};
return View_App0
})
```

4.0

```
(function anonymous(jit_createRendererType2_0,jit_viewDef_1,jit_textDef_2,jit_elementDef_3) {
  var styles_App = [];
  var RenderType_App = jit_createRendererType2_0({encapsulation:2,styles:styles_App,
    data:{}});
  function View_App_0(_l) {
    return jit_viewDef_1(0,[
      (_l()),jit_textDef_2(-1,null,['\\n  ']),
      (_l()),jit_elementDef_3(1,0,null,null,4,'div',[],null,null,null,null),
      (_l()),jit_textDef_2(-1,null,['\\n  ']),
      (_l()),jit_elementDef_3(3,0,null,null,1,'h2',[],null,null,null,null),
      (_l()),jit_textDef_2(4,null,['Hello ','']),
      (_l()),jit_textDef_2(-1,null,['\\n  ']),
      (_l()),jit_textDef_2(-1,null,['\\n  '])
    ],
    null,
    function(_ck, _v) {
      var _co = _v.component;
      var currVal_0 = _co.name;
      _ck(_v,4,0,currVal_0);
    });
    return {RenderType_App:RenderType_App,View_App_0:View_App_0};
  }
})
```

6.0

```
(function anonymous(i0) {
  var App = (function () {
    function App() {}
    App.ngComponentDef = i0.eDefineComponent({
      type: App,
      selectors: [['my-app']],
      factory: function App_Factory() {
        return new App();
      },
      template: function App_Template(ctx, cm) {
        if (cm) {
          i0.eE(0, "div");
          i0.eE(1, "h2");
          i0.eT(2);
          i0.ee();
          i0.ee();
        }
        i0.eT(2, i0.ei1('Hello ', ctx.name, ""));
      }
    });
    return App;
  })();
  i0.eRenderComponent(App);
})
```

9.0.0-rc.3

```
defineComponent({
  type: AppComponent,
  selectors: [
    ["app-root"]
  ],
  decls: 3,
  vars: 1,
  template: function AppComponent_Template(rf, ctx) {
    if (rf & 1) {
      elementStart(0, "div");
      elementStart(1, "h2");
      text(2);
      elementEnd();
      elementEnd();
    }
    if (rf & 2) {
      advance(2);
      textInterpolate1("Hello ", ctx.name, "");
    }
  },
  encapsulation: 2
});
```

1170 KB



# Bundle Size After Compiler

4.0

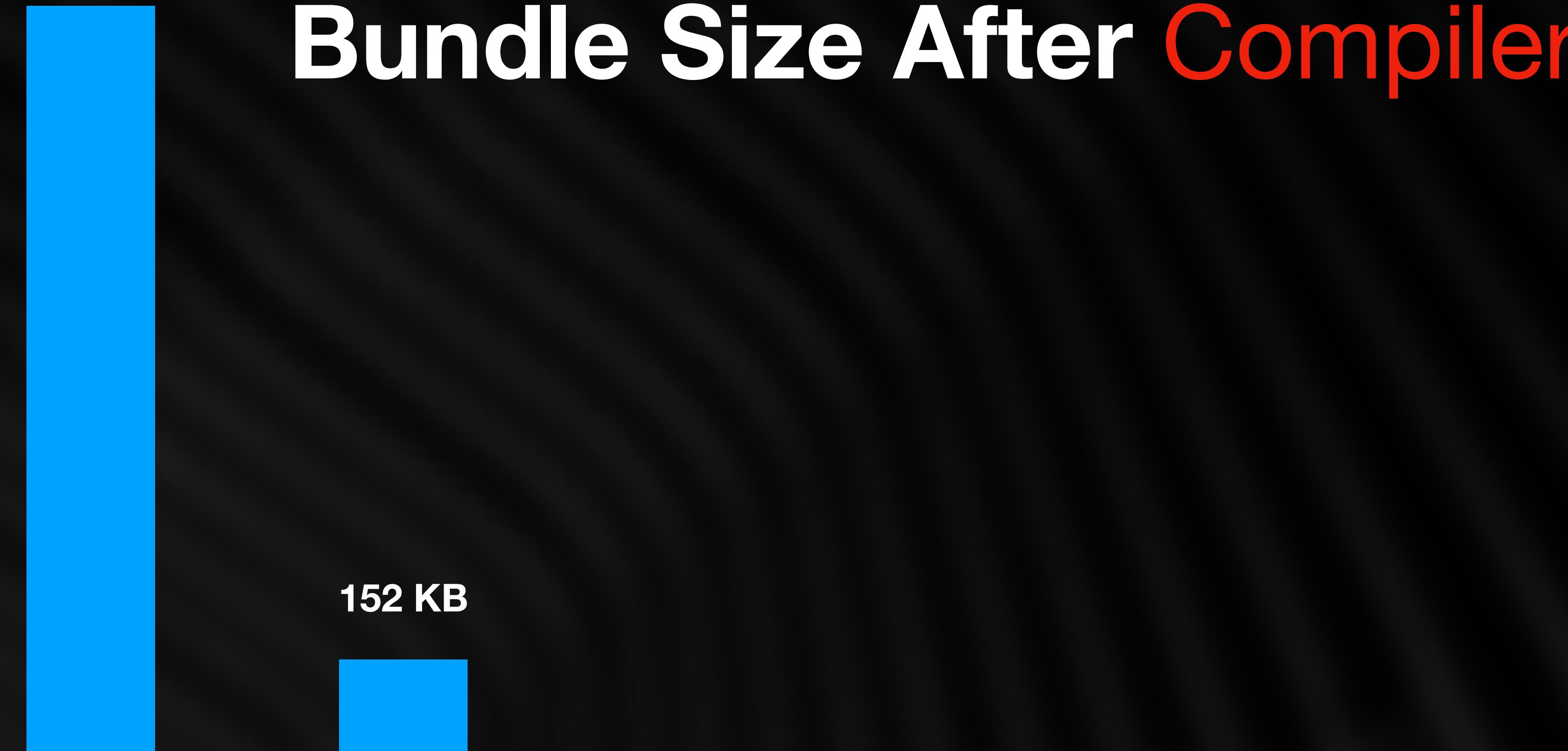
# Bundle Size After Compiler

1170 KB

152 KB

4.0

5.0



1170 KB

# Bundle Size After Compiler

Differential  
Loading

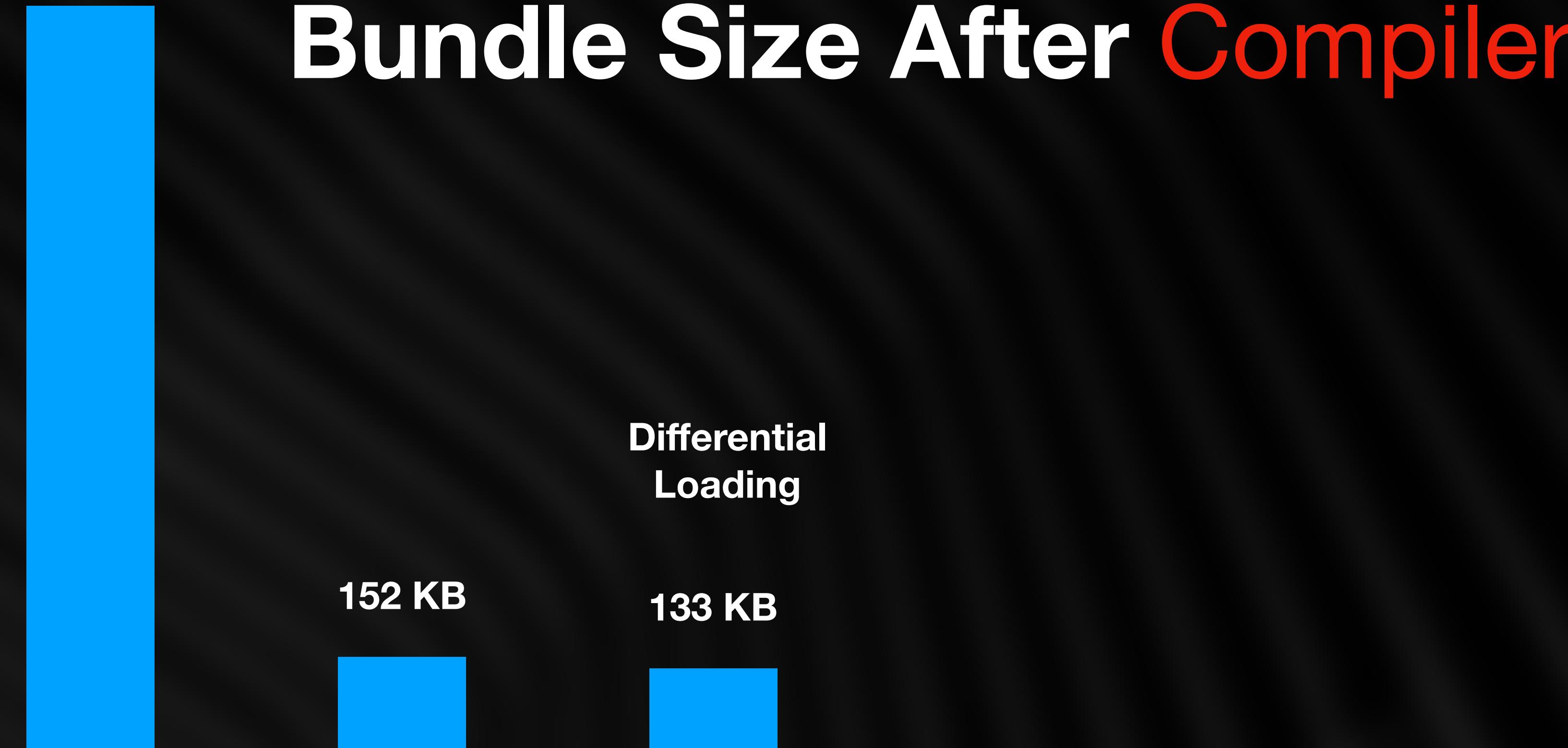
152 KB

133 KB

4.0

5.0

8.0



1170 KB

# Bundle Size After Compiler

Differential  
Loading

Ivy

152 KB

133 KB

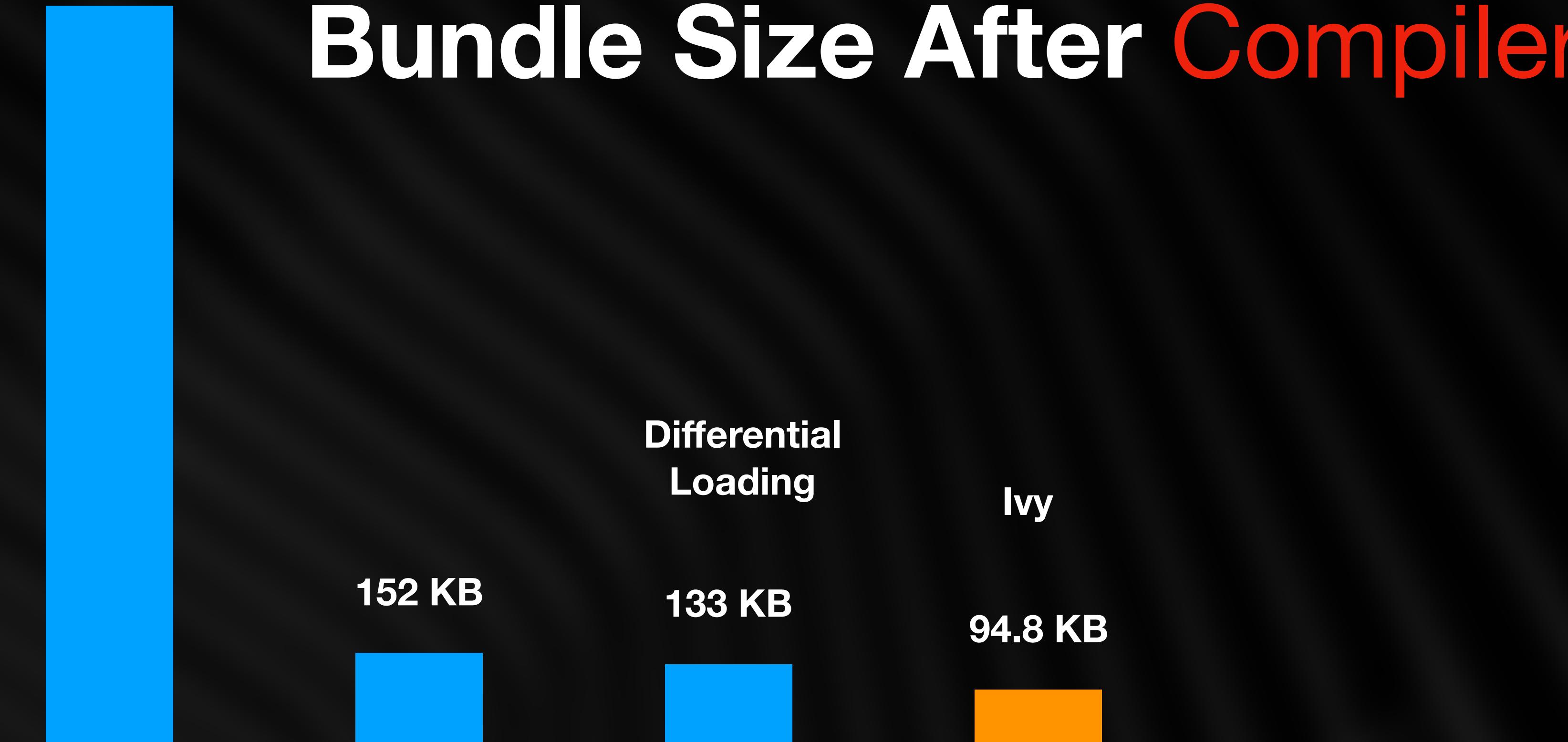
94.8 KB

4.0

5.0

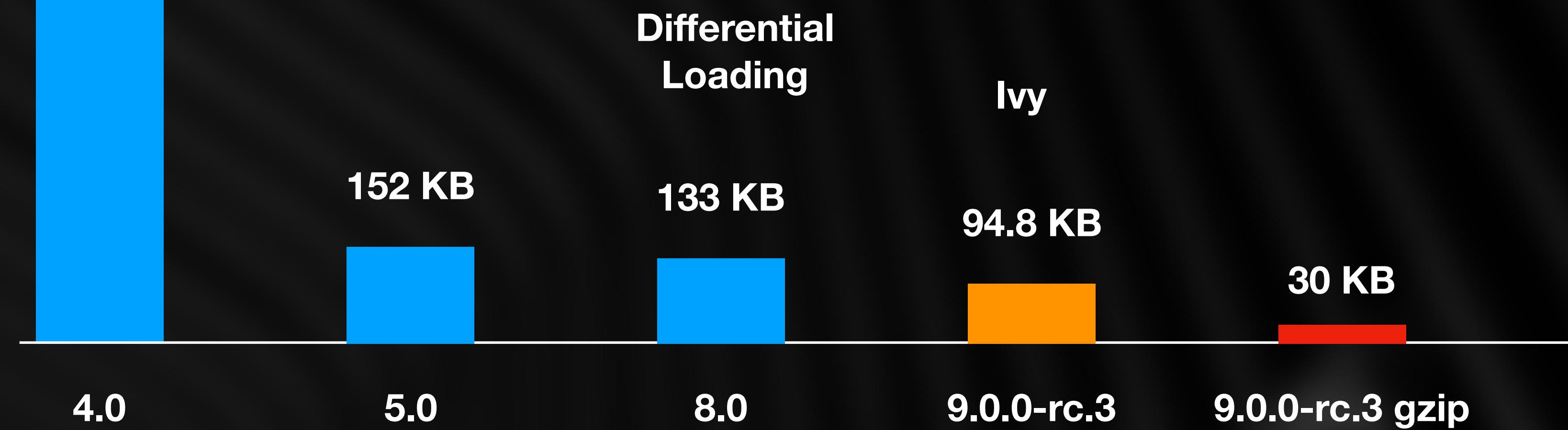
8.0

9.0.0-rc.3



1170 KB

# Bundle Size After Compiler



# Command-Line Interface

@angular/cli



Broccoli

Webpack

Bazel

# Stable & Reliable



Angular



Protractor



Forms



PWA



Augury



Language Services



Router



Elements



CDK



Universal



Karma



Labs



Compiler



i18n



Http



Material



Animations



CLI

# Release schedule & Semantic Versioning

DATE	STABLE RELEASE	COMPATIBILITY
October/November 2019	9.0.0	<sup>^</sup> 8.0.0
May 2020	10.0.0	<sup>^</sup> 9.0.0

# ng update & Update Guide

ng update @angular/cli @angular/core



One framework.  
Mobile & desktop.

[GET STARTED](#)



## DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

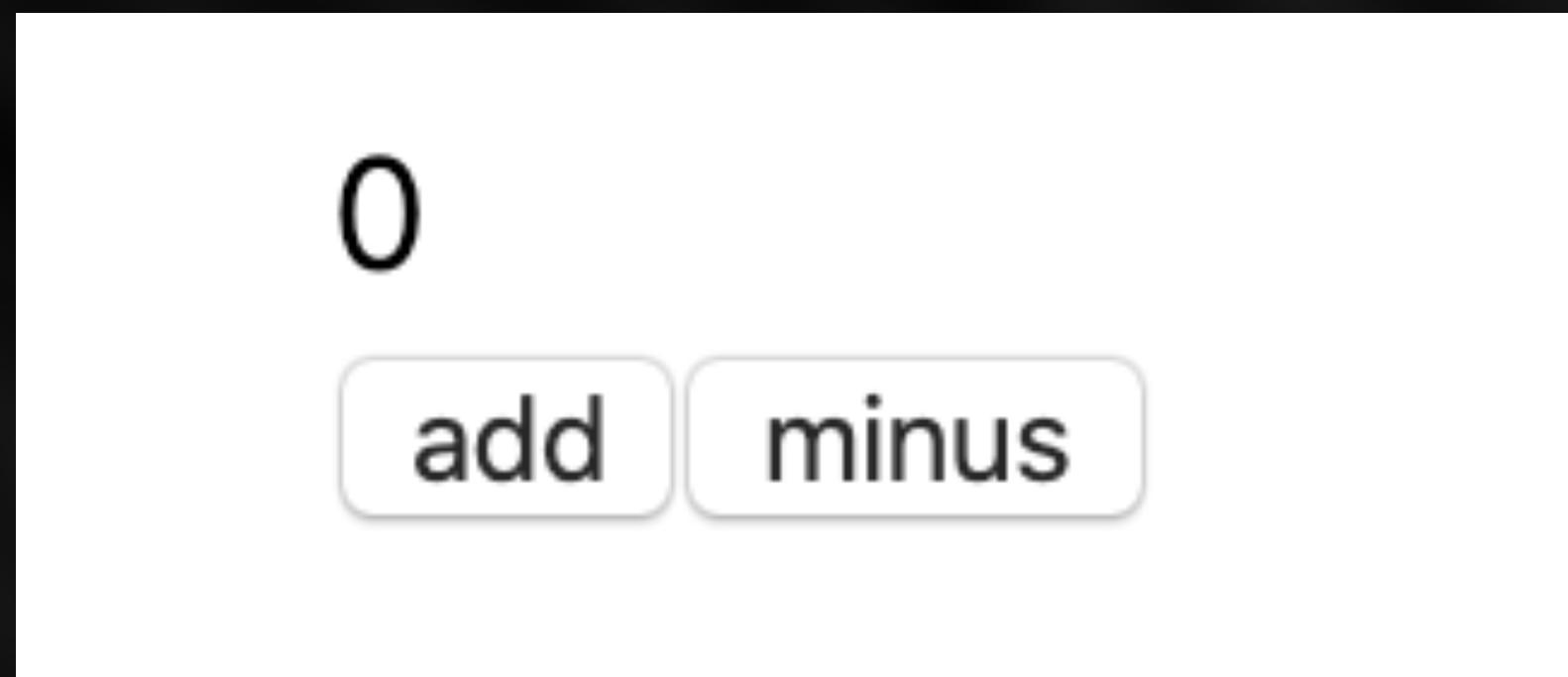


# The Hidden Docs in Angular

# The Hidden Docs in

# The Hidden Docs in Change Detection

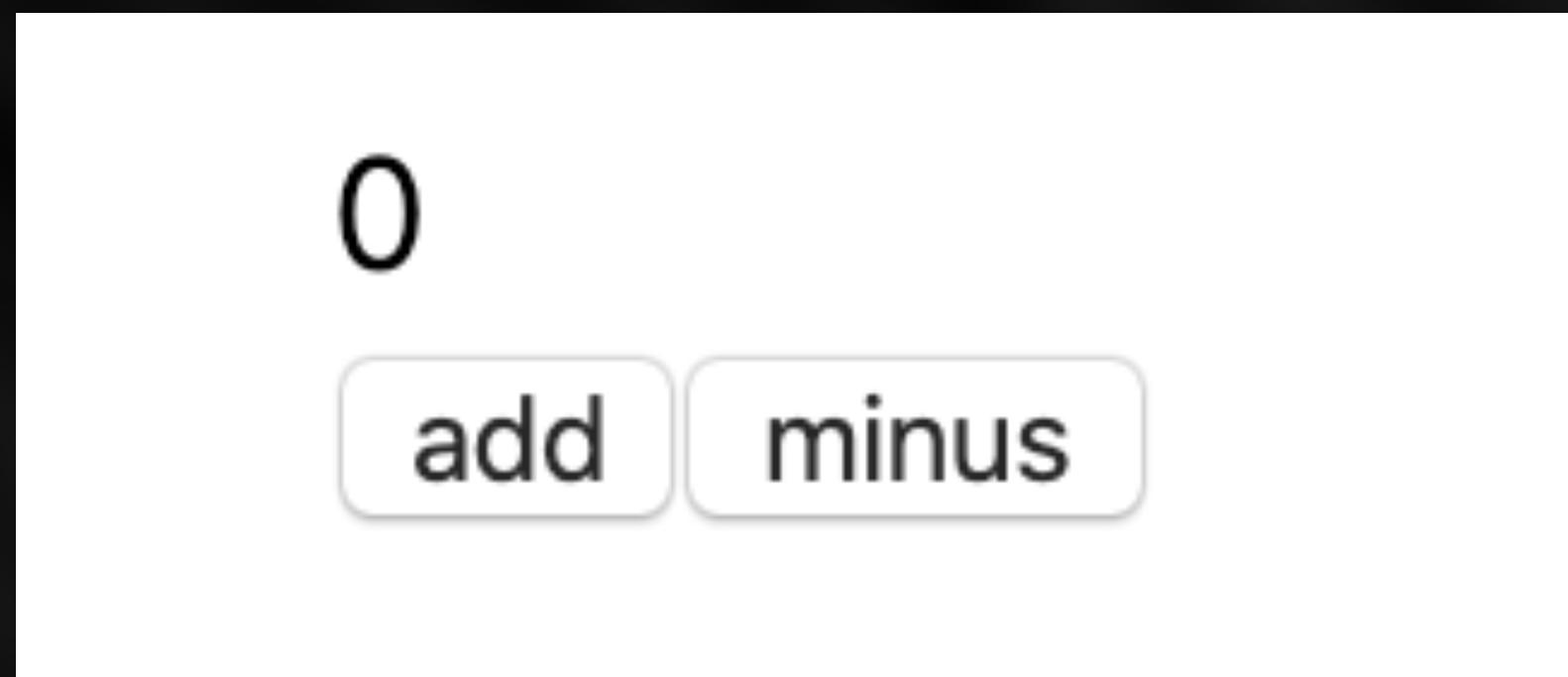
# Counter



```
@Component({
  selector: "hello",
  template: `
    <div>{{ count }}</div>
    <button (click)="add()">add</button>
    <button (click)="minus()">minus</button>
  `
})
export class HelloComponent {
  count = 0;
  add() {
    this.count += 1;
  }
  minus() {
    this.count -= 1;
  }
}
```

[DEMO](#)

# Counter



## Change Detection

```
@Component({
  selector: "hello",
  template: `
    <div>{{ count }}</div>
    <button (click)="add()">add</button>
    <button (click)="minus()">minus</button>
  `
})
export class HelloComponent {
  count = 0;
  add() {
    this.count += 1;
  }
  minus() {
    this.count -= 1;
  }
}
```

[DEMO](#)



# Change Detection

# Change Detection

## React

Change to **state** cause change detection

setState or hooks

# Change Detection

## React

Change to **state** cause change detection

setState or hooks

## Angular

?

# When to Trigger Change Detection?

## Browser Async API

- **Events:** all browser events (click, mouseover, keyup, etc.)
- **Timers:** setTimeout() and setInterval()
- **XHR:** Ajax requests

# When to Trigger Change Detection?



**zone.js**

Browser Async API  
[DEMO](#)

## Monkey Patch

- **Events:** all browser events (click, mouseover, keyup, etc.)
- **Timers:** setTimeout() and setInterval()
- **XHR:** Ajax requests

# Explore the Source Code

```
this._zone.onMicrotaskEmpty.subscribe(  
{  
  next: () =>  
    this._zone.run(() => {  
      this.tick();  
    });  
  }  
});
```

ApplicationRef

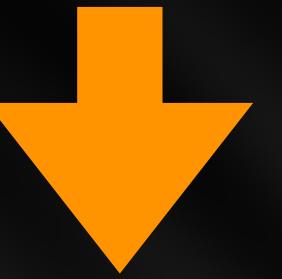
| this.applicationRef.tick()

```
_loadComponent(componentRef) {  
  this.tick();  
}
```

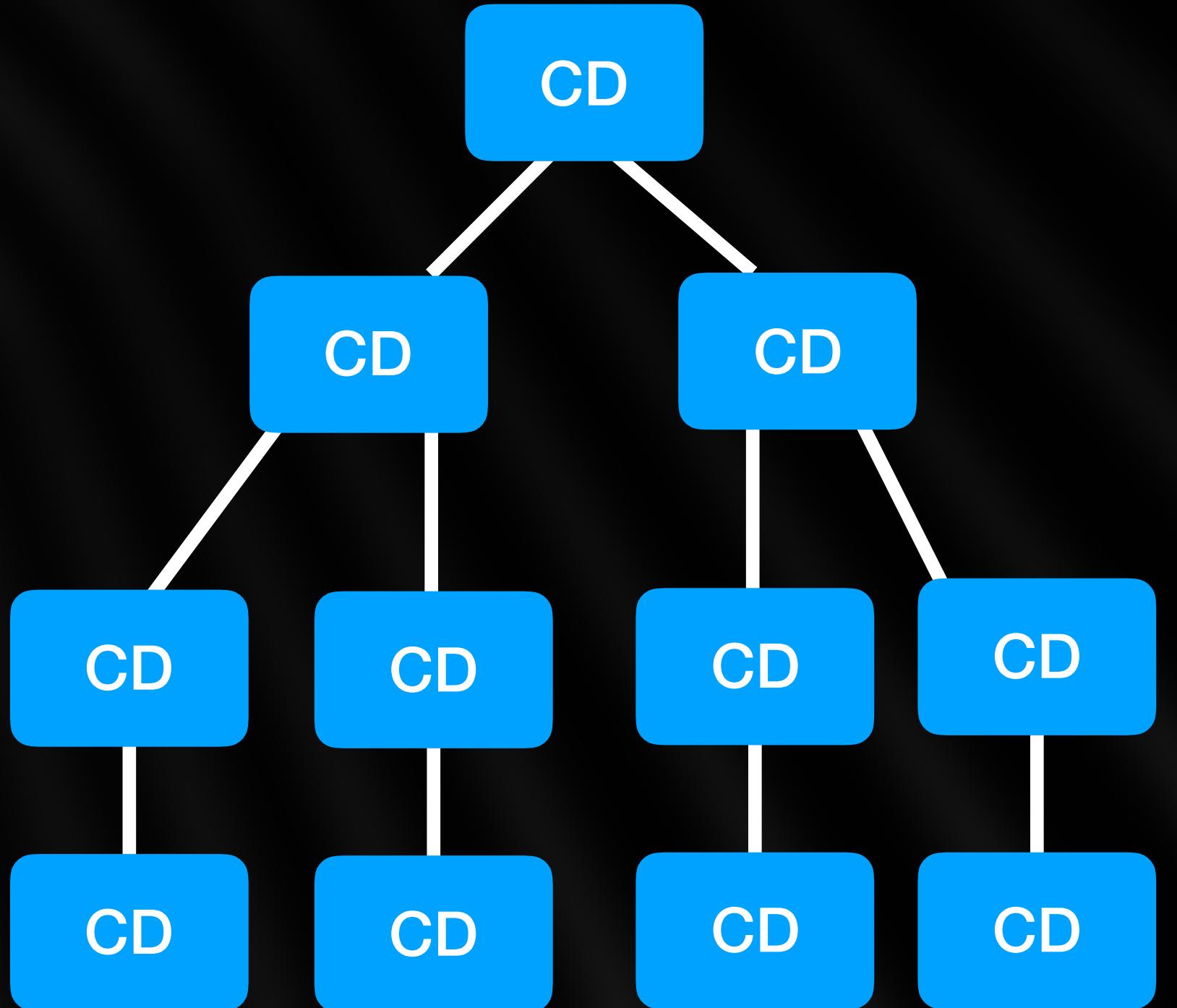
[Demo with zone.js](#)

# Explore the Source Code

```
| this.applicationRef.tick()
```



```
for (let view of this._views) {  
  view.detectChanges();  
}
```



ChangeDetectionRef

this.cdr.detectChanges()

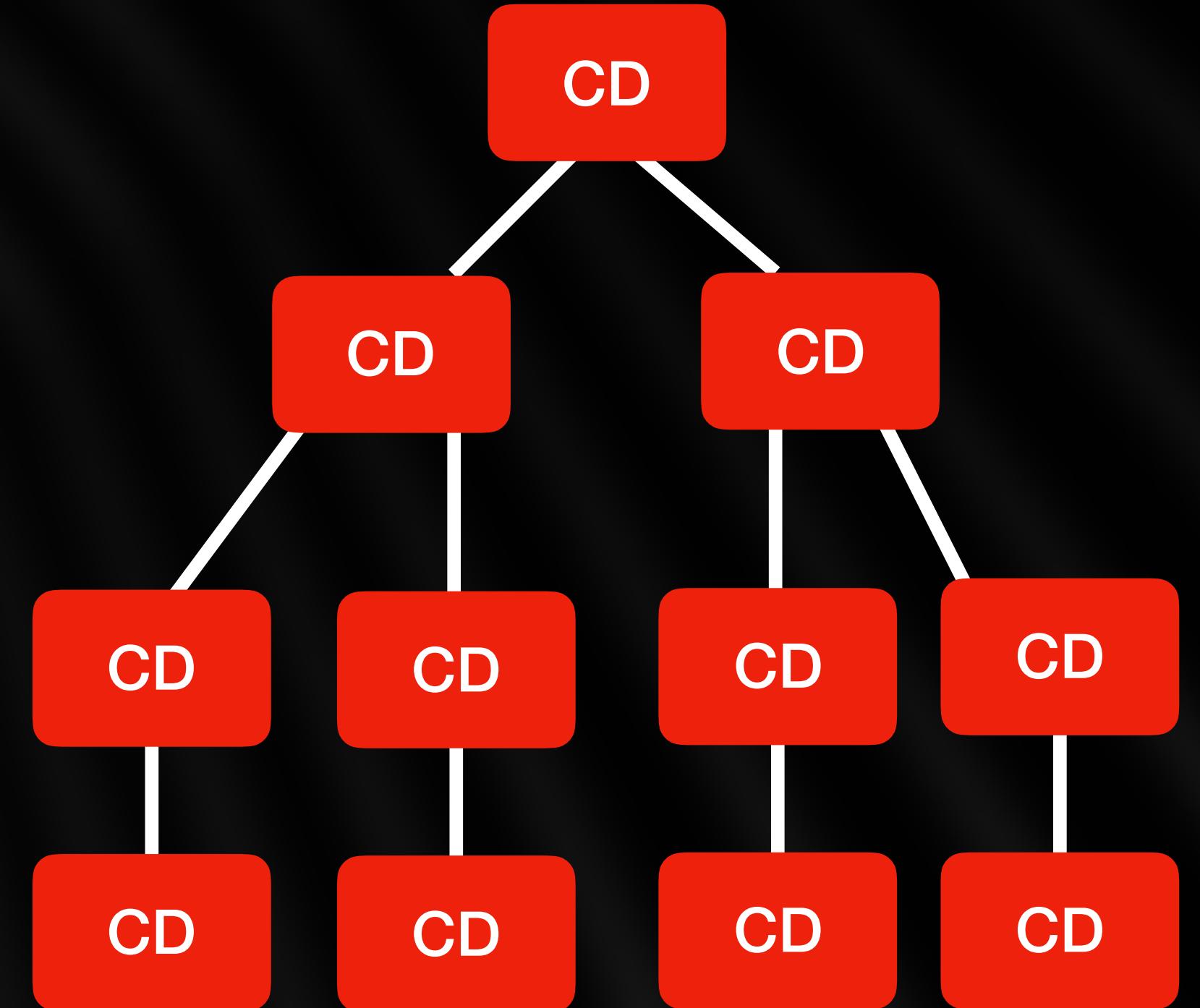
# Explore the Source Code

```
| this.applicationRef.tick()
```



```
for (let view of this._views) {  
  view.detectChanges();  
}
```

ChangeDetectionRef



[Demo](#) with ChangeDetectionRef

# Explore the Source Code

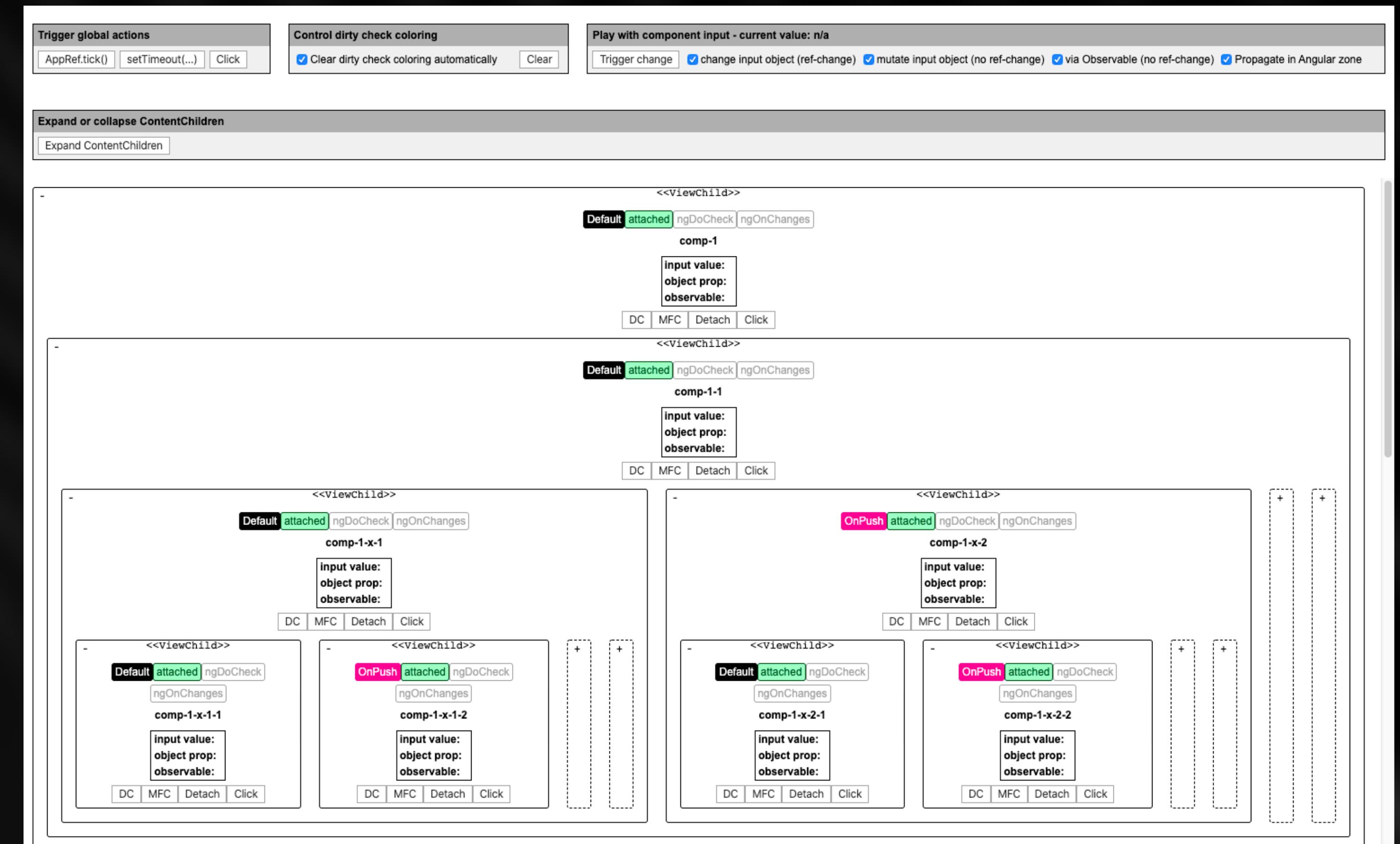
ChangeDetectionRef

markForCheck()

detectChanges()

detach()

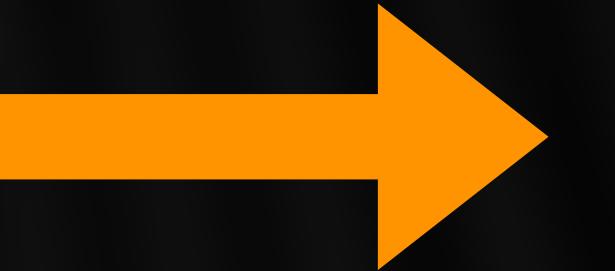
OnPush



# Detection Sequence

Why is it important?

ViewModel



View

DEMO

Error:

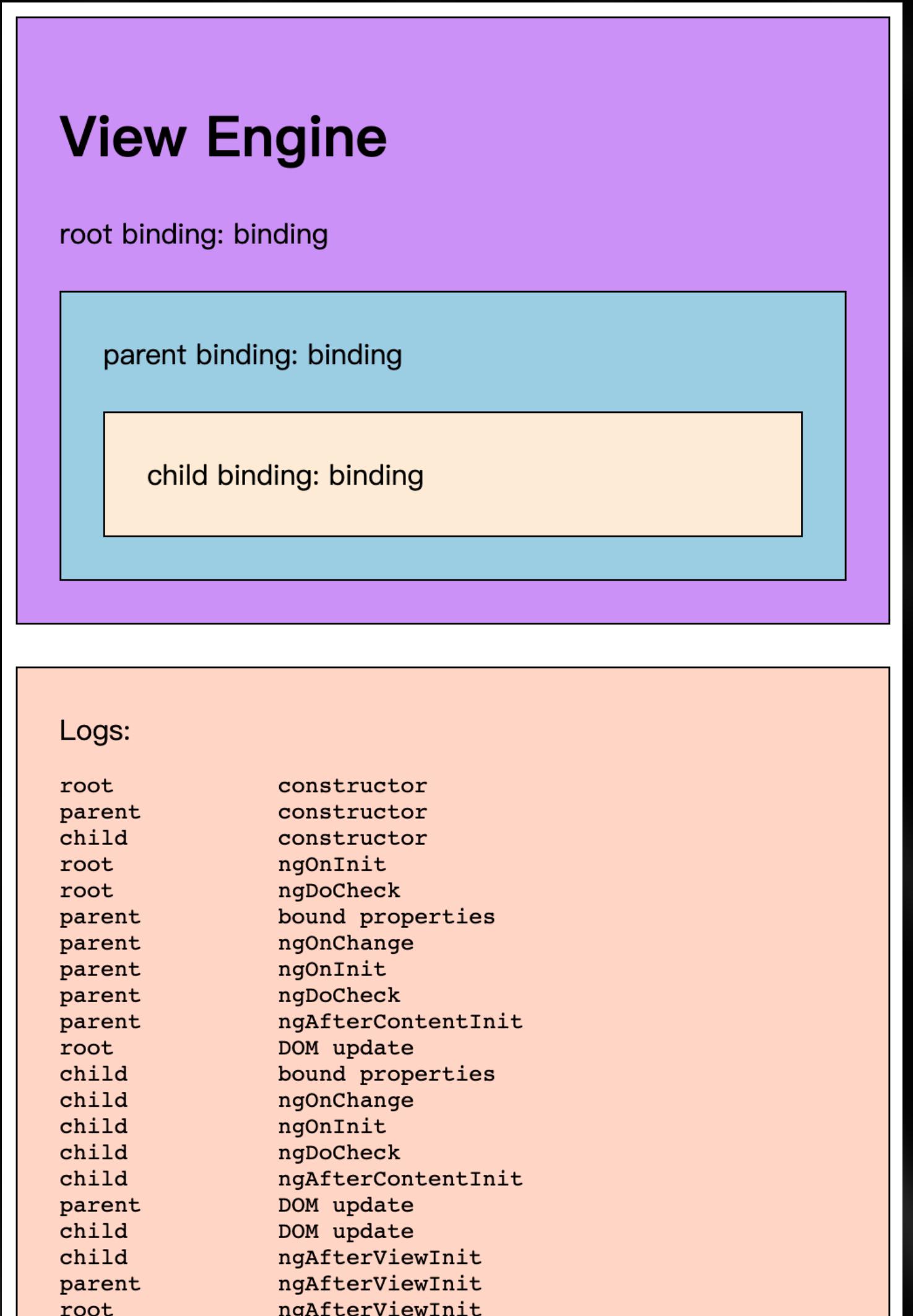
ExpressionChangedAfterItHasBeenCheckedError:  
Expression has changed after it was checked.

DEMO

# Detection Sequence

1. update bound properties for all child components
2. call OnChanges, OnInit, DoCheck and AfterContentInit lifecycle hooks on all child components
3. update DOM for the current component
4. run change detection for a child component
5. call ngAfterViewInit lifecycle hook for all child components

Sequence is slightly different under Ivy engine



# One Way Data Flow

vs

# Unidirectional Data Flow

# Summary

1. When
2. Source Code
3. Detection Sequence

# Summary

1. When **zone.js + tick**
2. Source Code **ApplicationRef + ChangeDetectionRef**
3. Detection Sequence **ViewModel → View & Unidirectional Data Flow**

# The Hidden Docs in

# The Hidden Docs in Component | Directive

1. Selector
2. @Input
3. Content Projection
4. Dynamic Component

# Selector

# Explore the Source Code

## Selector

```
const _SELECTOR_REGEXP = new RegExp(  
  '(\\":not\\()|' +      //":not("'  
  '(-\\w+)|' +        // "tag"  
  '(?:\\.(\\w+))|' + // ".class"  
  // "-" should appear first in the regexp below as FF31 parses "[.-\\w]" as a range  
  '(?:\\[([-\\w*]+)(?:=(["\\"]?)([^\\]\\\"\\"]*)\\5)?\\])|' + // "[name]", "[name=value]",  
    // "[name=\"value\"]",  
    // "[name='value']"  
    // ")"  
    // ","  
  '(\\))|' +  
  '(\\s*,\\s*)',  
  'g');
```

1. :not()
2. Tag
3. Attribute

export interface Component extends Directive

Same Selector

# Attribute Selector in Component

# Attribute Selector in Component

## 1. Supporting Original Attributes

```
<button ngx-button disabled>{{name}}</button>
```

**VS**

```
<ngx-button ngxDisabled></ngx-button>
```

# Attribute Selector in Component

## 1. Supporting Original Attributes

```
<button ngx-button disabled>{{name}}</button>
```

VS

```
<ngx-button ngxDisabled></ngx-button>
```

## 2. Keep DOM structure

```
<div class="parent">
  <div class="child" ngx-attribute></div>
</div>
```

```
.parent > .child{
  height: 200px;
  background: gray;
}
```

Tag Selector in  
Directive

Lighthouse Scoring

# Tag Selector in Directive

## Lighthouse Scoring

```
import { Directive, HostBinding } from '@angular/core';

@Directive({
  selector: 'a[target="_blank"]:not([rel="noopener"])'
})
export class SafeLinkDirective {
  @HostBinding('style.color') color = 'red';
  @HostBinding('rel') rel = 'noopener';
}
```

@Input

# typeof @Input data

```
<check [data]="true"></check>
<check bind-data="true"></check>
<check data="true"></check>
<check data="{{true}}></check>
```

# typeof @Input data

```
<check [data]="true"></check>
<check bind-data="true"></check>
<check data="true"></check>
<check data="{{true}}></check>
```

Input type: boolean  
Input type: boolean  
Input type: string  
Input type: string

# typeof @Input data

```
<check [data]="true"></check>
<check bind-data="true"></check>
<check data="true"></check>
<check data="{{true}}></check>
```

```
this.data === true
```

Input type: boolean  
Input type: boolean  
Input type: string  
Input type: string

# typeof @Input data

```
<check [data]="true"></check>
<check bind-data="true"></check>
<check data="true"></check>
<check data="{{true}}></check>
```

```
this.data === true
```

Input type: boolean  
Input type: boolean  
Input type: string  
Input type: string

## Input Correction

# @Input as Attribute

```
<button disabled></button>
<button [disabled]="true"></button>
<button [disabled]="false"></button>
```

# @Input as Attribute

```
<button disabled></button>
<button [disabled]="true"></button>
<button [disabled]="false"></button>
```

```
<ngx-attribute [ngxOpen]="true"></ngx-attribute>
<ngx-attribute [ngxOpen]="false"></ngx-attribute>
<ngx-attribute ngxOpen></ngx-attribute>
```

# @Input as Attribute

```
<button disabled></button>
<button [disabled]="true"></button>
<button [disabled]="false"></button>
```

```
<ngx-attribute [ngxOpen]="true"></ngx-attribute>
<ngx-attribute [ngxOpen]="false"></ngx-attribute>
<ngx-attribute ngxOpen></ngx-attribute>
```

```
@Input()
set ngxOpen(value) {
  this._open = this.coerceBooleanProperty(value);
}
get ngxOpen() {
  return this._open;
}
coerceBooleanProperty(value: any): boolean {
  return value != null && `${value}` !== "false";
}
```

# @Input as Attribute

```
<button disabled></button>
<button [disabled]="true"></button>
<button [disabled]="false"></button>
```

```
<ngx-attribute [ngxOpen]="true"></ngx-attribute>
<ngx-attribute [ngxOpen]="false"></ngx-attribute>
<ngx-attribute ngxOpen></ngx-attribute>
```

```
@Input()
set ngxOpen(value) {
  this._open = this.coerceBooleanProperty(value);
}
get ngxOpen() {
  return this._open;
}
coerceBooleanProperty(value: any): boolean {
  return value != null && `${value}` !== "false";
}
```

## Decorators

```
@InputBoolean() ngxOpen = false;
```

@Input with \*

# @Input with \*

```
<div *ngIf="heroes else loading">  
  ...  
</div>  
<ng-template #loading>  
  <div>Loading...</div>  
</ng-template>
```

# @Input with \*

```
<div *ngIf="heroes else loading">
  ...
</div>
<ng-template #loading>
  <div>Loading...</div>
</ng-template>
```

```
@Input()
set ngIfElse(templateRef: TemplateRef<NgIfContext>|null) {
  assertTemplate('ngIfElse', templateRef);
  this._elseTemplateRef = templateRef;
  this._elseViewRef = null; // clear previous view if any.
  this._updateView();
}
```

DEMO

# Content Projection

# Content Projection

## Multi-Slot

```
<ng-content select="slot-name"></ng-content>
```

# Content Projection

## Multi-Slot

```
<ng-content select="slot-name"></ng-content>
```

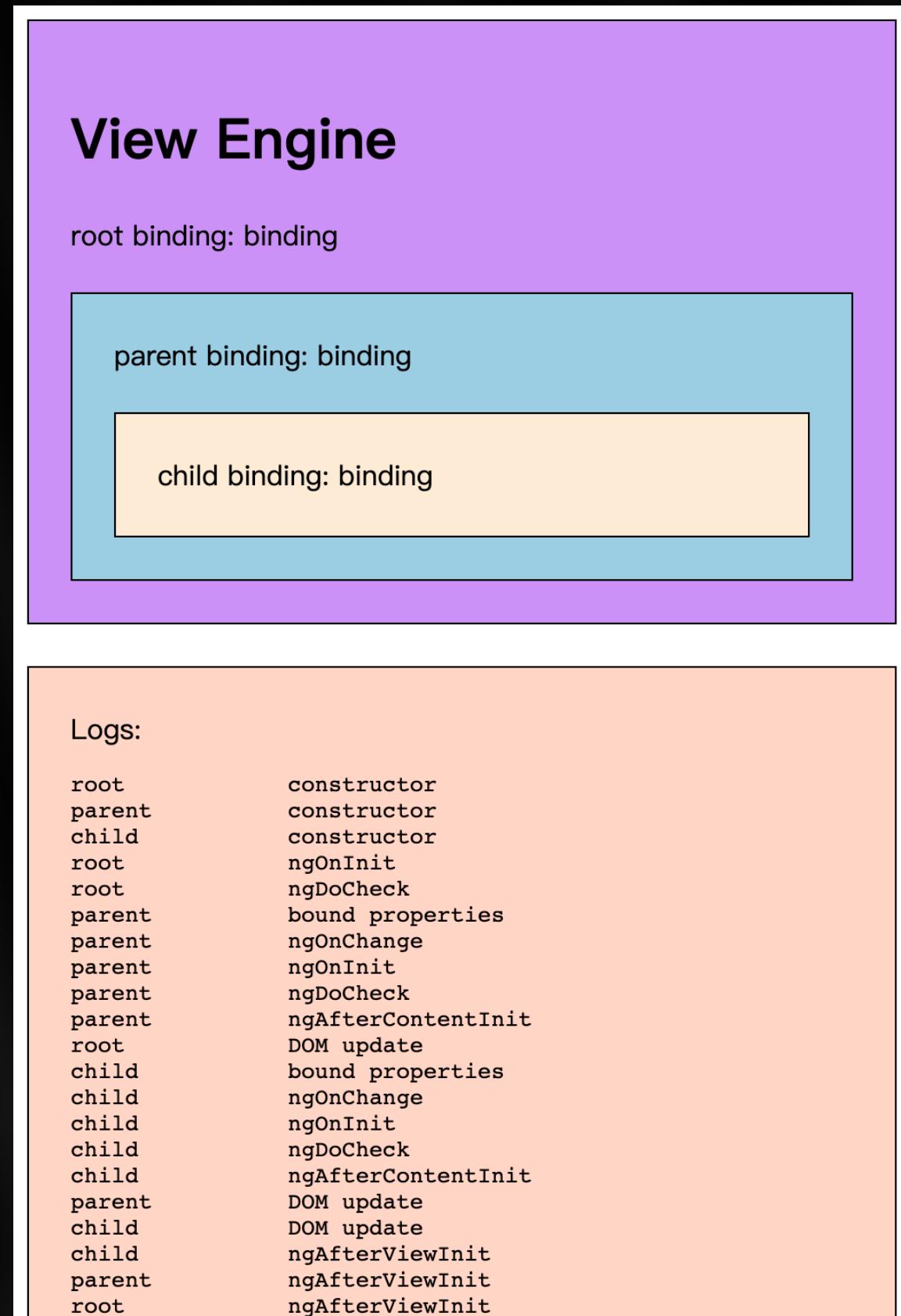
1. Tag Selector
2. Attribute Selector
3. Not Selector
4. Default Selector
5. ngProjectAs

# Content Projection

## Component Interacting

### ContentChildren

1. ChangeDetection Error
2. NgTemplateOutlet



# Content Projection

## Component Interacting

### Dependency Injection

1. NgTemplateOutlet
2. Get Root Component

```
<app-dropdown>
  <app-menu>
    <app-menu-item></app-menu-item>
    <app-menu-item></app-menu-item>
    <app-menu-item></app-menu-item>
  </app-menu>
</app-dropdown>

<app-menu>
  <app-menu-item></app-menu-item>
  <app-menu-item></app-menu-item>
  <app-menu-item></app-menu-item>
</app-menu>

<app-menu-item></app-menu-item>
<app-menu-item></app-menu-item>
<app-menu-item></app-menu-item>
```

dropdown item  
dropdown item  
dropdown item

menu item  
menu item  
menu item

base item  
base item  
base item

# Content Projection

## Content Empty Detection

# Content Projection

## Content Empty Detection

```
<nz-badge [nzCount]="25"></nz-badge>
```



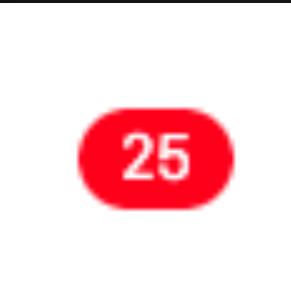
# Content Projection

## Content Empty Detection

```
<nz-badge [nzCount]="25"></nz-badge>
```



```
<nz-badge [nzCount]="5">  
  <a class="head-example"></a>  
</nz-badge>
```



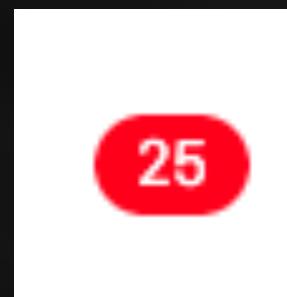
# Content Projection

## Content Empty Detection

```
<nz-badge [nzCount]="25"></nz-badge>
```



```
<nz-badge [nzCount]="5">  
  <a class="head-example"></a>  
</nz-badge>
```



No ability to check if `<ng-content>` was provided to the component #26083

[! Open](#) sharikovvladislav opened this issue on 24 Sep 2018 · 16 comments



sharikovvladislav commented on 24 Sep 2018 · edited

+ ...

I'm submitting a...

- Regression (a behavior that used to work and stopped working in a new release)
- Bug report
- Performance issue
- Feature request
- Documentation issue or request
- Support request => Please do not submit support request here, instead see <https://g>:
- Other... Please describe:

### Current behavior

We can use `ng-content` in our components. But we don't have any options out of the box to check if anything is passed as `ng-content` or not. Actually, was anything passed between `>` and `</` of our component usage in HTML.

We can do this by adding some wrap component (and if I understood correctly you can not use stuff like `ng-container` there). Like `div`. And you put `ng-content` to it. You can get `ref` for this `div` and check if this `div` has children. If yes - `ng-content` was provided, otherwise - no `ng-content`. Huge minus of this approach is you have to create some additional entity to check it.

### Expected behavior

#### Assignees

No one assigned

#### Labels

`comp: core`

`type: feature`

#### Projects

None yet

#### Milestone

Backlog

#### Notifications

Customize

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

#### 11 participants



# Content Projection

## Content Empty Detection

No ability to check if `<ng-content>` was provided to the component #26083

[Open](#) sharikovvladislav opened this issue on 24 Sep 2018 · 16 comments

sharikovvladislav commented on 24 Sep 2018 · edited

I'm submitting a...

[ ] Regression (a behavior that used to work and stopped working in a new release)  
[ ] Bug report  
[ ] Performance issue  
[x] Feature request  
[ ] Documentation issue or request  
[ ] Support request => Please do not submit support request here, instead see <https://g>:  
[ ] Other... Please describe:

**Current behavior**

We can use `ng-content` in our components. But we don't have any options out of the box to check if anything is passed as `ng-content` or not. Actually, was anything passed between `>` and `</` of our component usage in HTML.

We can do this by adding some wrap component (and if I understood correctly you can not use stuff like `ng-container` there). Like `div`. And you put `ng-content` to it. You can get ref for this `div` and check if this `div` has children. If yes - `ng-content` was provided, otherwise - no `ng-content`. Huge minus of this approach is you have to create some additional entity to check it.

**Expected behavior**

New issue

Assignees  
No one assigned

Labels  
`comp: core` (selected)  
`type: feature`

Projects  
None yet

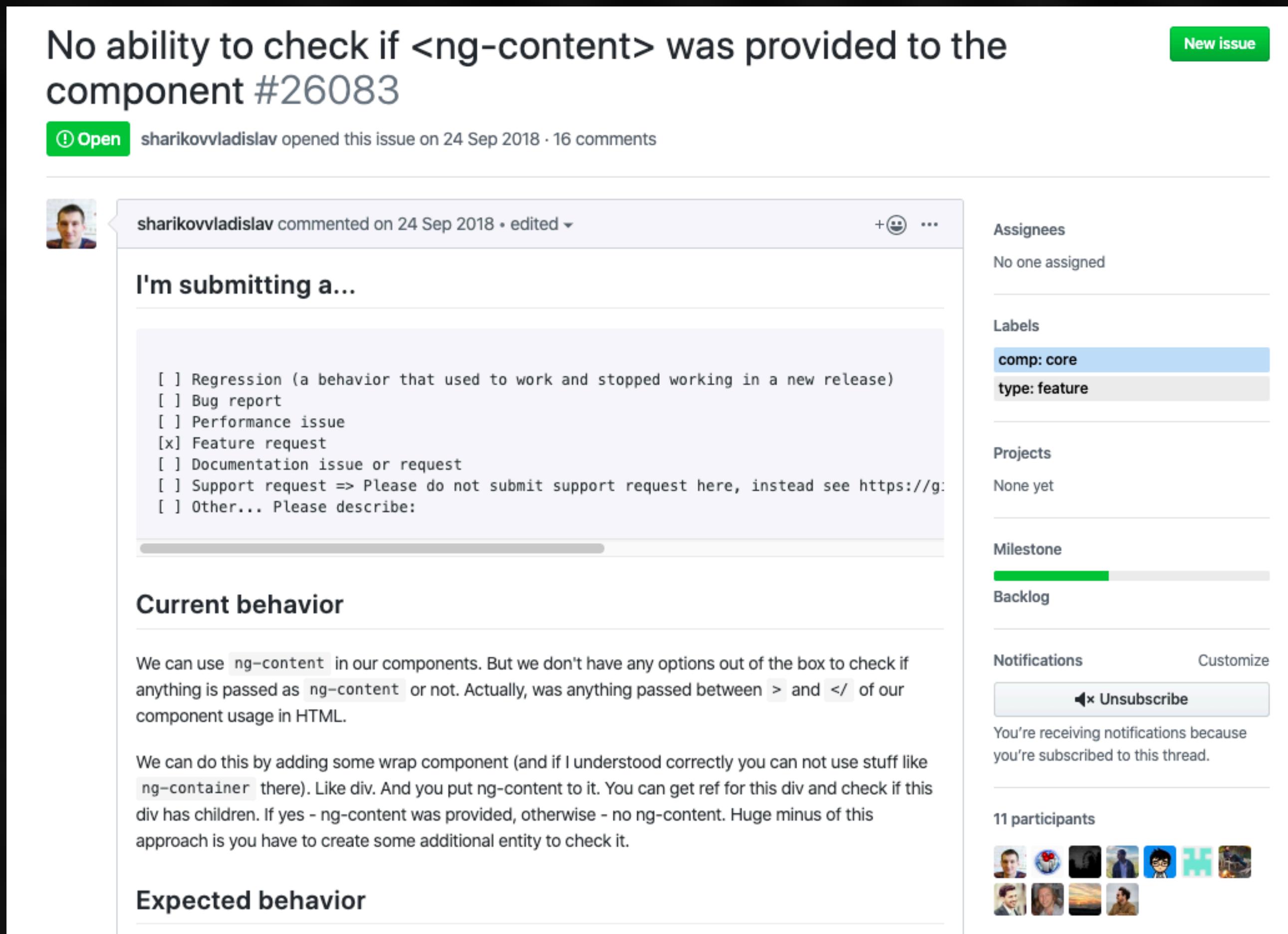
Milestone  
Backlog

Notifications  
Unsubscribe

You're receiving notifications because you're subscribed to this thread.

Customize

11 participants



## MutationObserver

## ContentObserver

## Demo

# Content Projection

## Copy Content

```
<copy [copy] = "8">
  <button>copy me</button>
</copy>
```

# Content Projection

## Copy Content

```
<copy [copy]="8">  
  <button>copy me</button>  
</copy>
```



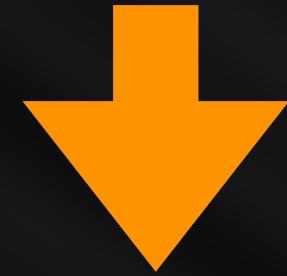
```
<button>copy me</button>
```

X 8

# Content Projection

## Copy Content

```
<copy [copy]="8">  
  <button>copy me</button>  
</copy>
```



```
<button>copy me</button>
```

X 8

### provide method to multiple copies of ng-content #21462

 **Closed** vthinkxie opened this issue on 11 Jan 2018 · 6 comments



vthinkxie commented on 11 Jan 2018 · edited

Contributor ...

I'm submitting a...

- Regression (a behavior that used to work and stopped working in a new release)
- Bug report
- Feature request
- Documentation issue or request
- Support request => Please do not submit support request here, instead see <https://g>:

#### Current behavior

Is there any way to multiple copies of `ng-content`?  
I try to copy it with the demo below, but not work.

Only `ContentChild` can be copied, but it will bring much trouble when I want to build a `lib` for users that they have to wrap all the content with `ng-template` everywhere.

Here is the mini reproduction link:

<https://stackblitz.com/edit/angular-mofgqd?file=app%2Fchild.component.ts>

There is similar code in Material Design [here](#), but they only need to move the `ng-content` to right place other than copy it.

Have I missed something, or is there any way to hack this?

#### Expected behavior

provide some methods to copy `ng-content` like add `copy` method like `insert` or some other way to implement this.

[angular/packages/core/src/linker/view\\_container\\_ref.ts](angular/packages/core/src/linker/view_container_ref.ts)  
Line 96 in 6be9c04

96     abstract insert(viewRef: ViewRef, index?: number): ViewRef;

Assignees  
No one assigned

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Notifications      Customize  
 Unsubscribe  
You're receiving notifications because you authored the thread.

4 participants



# Content Projection

## Copy Content

provide method to multiple copies of ng-content #21462

**Closed** vthinkxie opened this issue on 11 Jan 2018 · 6 comments

vthinkxie commented on 11 Jan 2018 · edited

I'm submitting a...

[ ] Regression (a behavior that used to work and stopped working in a new release)  
[ ] Bug report  
[x] Feature request  
[ ] Documentation issue or request  
[ ] Support request => Please do not submit support request here, instead see https://g...

**Current behavior**

Is there any way to multiple copies of ng-content ?  
I try to copy it with the demo below, but not work.

Only `ContentChild` can be copied, but it will bring much trouble when I want to build a `lib` for users that they have to wrap all the content with `ng-template` everywhere.

Here is the mini reproduction link:

<https://stackblitz.com/edit/angular-mofgqd?file=app%2Fchild.component.ts>

There is similar code in Material Design [here](#), but they only need to move the `ng-content` to right place other than copy it.

Have I missed something, or is there any way to hack this?

**Expected behavior**

provide some methods to copy `ng-content` like add `copy` method like `insert` or some other way to implement this.

`angular/packages/core/src/linker/view_container_ref.ts`  
Line 96 in 6be9c04

96     abstract insert(viewRef: ViewRef, index?: number): ViewRef;

<ng-template>

\*

Demo

# Content Projection

## Move Content

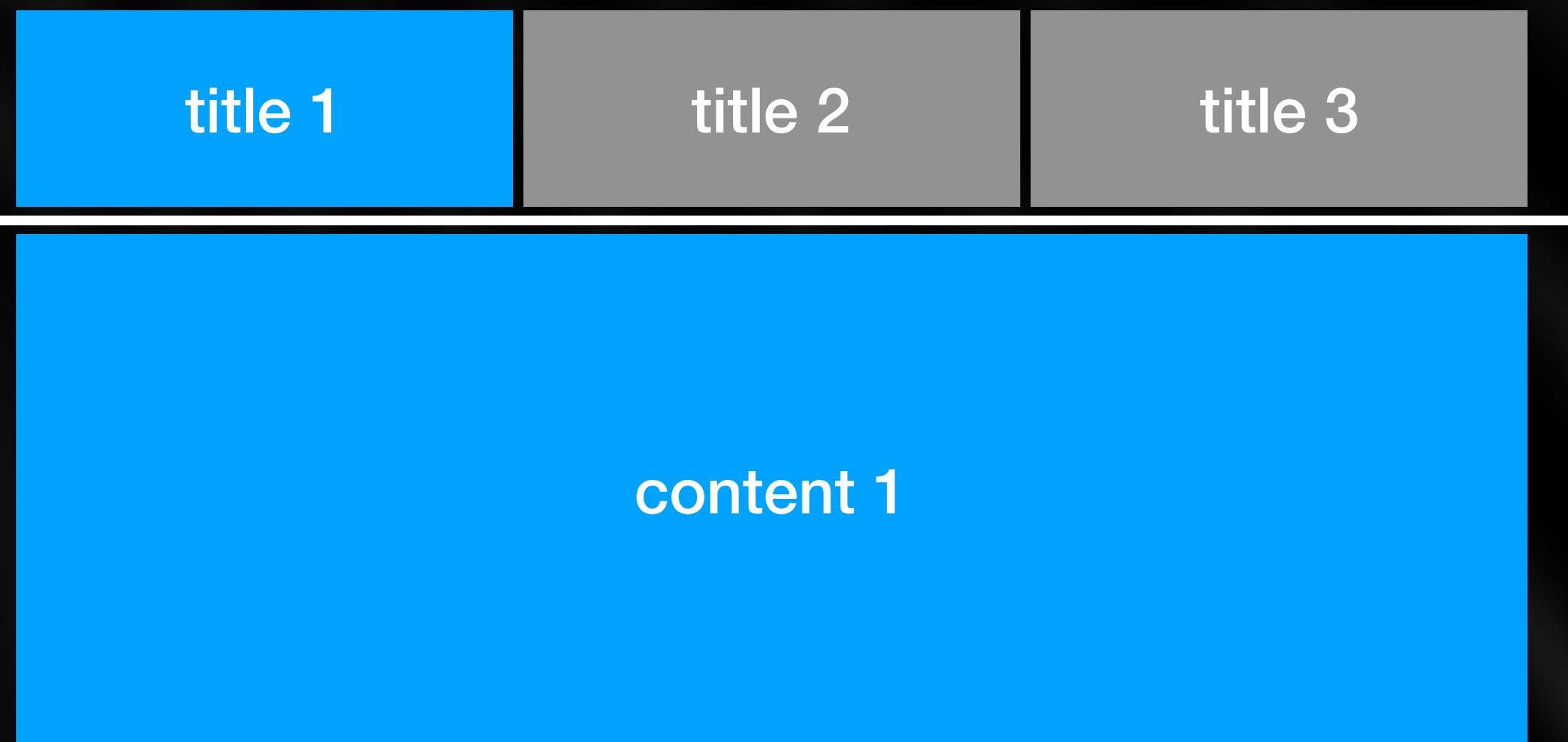
```
<tabset>
  <tab title="title 1">content 1</app-tab>
  <tab title="title 2">content 2</app-tab>
  <tab title="title 3">content 3</app-tab>
</tabset>
```



# Content Projection

## Move Content

```
<tabset>
  <tab title="title 1">content 1</app-tab>
  <tab title="title 2">content 2</app-tab>
  <tab title="title 3">content 3</app-tab>
</tabset>
```



# Content Projection

Move Content

```
<ng-template>  
  <ng-content></ng-content>  
</ng-template>
```

Eager Load

VS

Lazy Load

# Dynamic Component

# Dynamic Component

## The missing compile

Template String

compile

Dynamic Component

JitCompilerFactory

```
<input [(ngModel)]="context.name">
```

Load Via Jit Compiler

JitCompilerFactory

# Dynamic Component

## The missing compile

```
export function createCompiler(compilerFactory: CompilerFactory) {
  return compilerFactory.createCompiler([compilerOptions]);
}

providers: [
  { provide: COMPILER_OPTIONS, useValue: compilerOptions, multi: true },
  { provide: CompilerFactory, useClass: JitCompilerFactory, deps: [COMPILER_OPTIONS] },
  { provide: Compiler, useFactory: createCompiler, deps: [CompilerFactory] }
],
```

```
const dynamicComponent = Component({ template })(
  class {
    context = context;
  }
);
const dynamicModule = NgModule({
  declarations : [ dynamicComponent ],
  exports      : [ dynamicComponent ],
  entryComponents: [ dynamicComponent ],
  imports       : [ FormsModule ]
})(class DynamicModule {
});

this.compiledModule = await this.compiler.compileModuleAndAllComponentsAsync(dynamicModule);
```

DEMO

# Dynamic Component

## *Ivy Render*

```
import {  
  Component,  
  renderComponent as renderComponent,  
  Injector,  
  LifecycleHooksFeature as LifecycleHooksFeature  
} from "@angular/core";  
  
@Component({  
  selector : 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls : ['./app.component.css']  
})  
export class AppComponent {  
  
  constructor(private injector: Injector) {  
    import('./dynamic/dynamic.component').then(({ DynamicComponent }) => {  
      renderComponent(DynamicComponent, { injector, host: '#slot', hostFeatures: [LifecycleHooksFeature] });  
    });  
  }  
}
```

## Hidden Docs In Angular

	ChangeDetection	Directive + Component	Other
	<pre>// JS has NaN !== NaN export function looseIdentical(a: any, b: any): boolean {   return a === b    typeof a === 'number' &amp;&amp; typeof b === 'number' &amp;&amp; isNaN(a) &amp;&amp; isNaN(b); }</pre> <p>How  oldValue Vs new Value </p> <p>When  zone.js  Timers</p> <p>Events XHR</p> <pre>this._zone.onMicrotaskEmpty.subscribe(   {     next: () =&gt; {       this._zone.run(() =&gt; {         this.tick();       });     }   }); </pre> <p>zone.js </p> <p>ApplicationRef  load root component </p> <p>Source Code  for (let view of this._views) { view.detectChanges(); }</p> <p>tick </p> <p>markForCheck</p> <p>detectChanges</p> <p>detach</p> <p>OnPush</p> <p>Important: ViewModel &lt;-&gt; View</p> <p>Detection Sequence  1. update bound properties for all child components 2. call OnChanges, OnInit, DoCheck and AfterContentInit lifecycle hooks on all child components 3. update DOM for the current component 4. run change detection for a child component 5. call ngAfterViewInit lifecycle hook for all child components</p> <p>One Way Data Flow</p> <p>ExpressionChangedAfterItHasBeenCheckedError  Unidirectional Data Flow</p> <p>update 1st data in 2nd or update 3rd in 5th</p>	<p>selector  Attribute selector in component</p> <p>Tag selector in directive</p> <p>typeof @Input data  [data]="value" vs data="value" vs data="{{value}}"</p> <p>@Input  Input correction</p> <p>@Input as attribute  Decorator</p> <p>input with *  ngIf</p> <p>Tag Selector</p> <p>Multi-Slot  Attribute Selector</p> <p>Not Selector</p> <p>Default Selector</p> <p>ngProjectAs</p> <p>Content Projection  ChangeDetection Error</p> <p>Component Interacting  ContentChildren  NgTemplateOutlet</p> <p>Get Root Component  Di  NgTemplateOutlet</p> <p>Content Empty Detection  ContentObserver</p> <p>MutationObserver</p> <p>Copy Content  ng-template</p> <p>*</p> <p>ng-template</p> <p>Move Content  Eager Load</p> <p>LazyLoad</p> <p>Template string -&gt; Dynamic Component</p> <p>Ivy Render</p>	<p>Theme  https://ng.ant.design/blog/custom-theme/</p> <p>RouterModule  router data in easy way</p> <p>service provideIn  ServiceModule</p> <p>multiple project  relative path with path in tsconfig</p> <p>http client  dynamic interceptor according to header</p> <p>custom-builders  stylePreprocessorOptions</p> <p>angular.json configuration </p>

May 24, 2016 9:27am

执衡 authored 4 years ago

```
nz-checkbox.component.ts 2.19 KB
```

```
1 import {  
2   Component,  
3   OnInit,  
4   ViewEncapsulation,  
5   Input,  
6   Output,  
7   ElementRef,  
8   AfterContentInit,  
9   Renderer,  
10  HostListener,  
11  EventEmitter  
12 } from '@angular/core';  
13 import {NgClass} from '@angular/common';  
14 @Component({  
15   selector: '[nz-checkbox]',  
16   directives: [NgClass],  
17   encapsulation: ViewEncapsulation.None,
```

**ANGULAR ATTACK**

ENTRIES PRIZES RULES SPONSORS FAQ BLOG SIGN IN WITH GITHUB

## NAZA

135 commits 88 pushes 26 deploys 27 votes

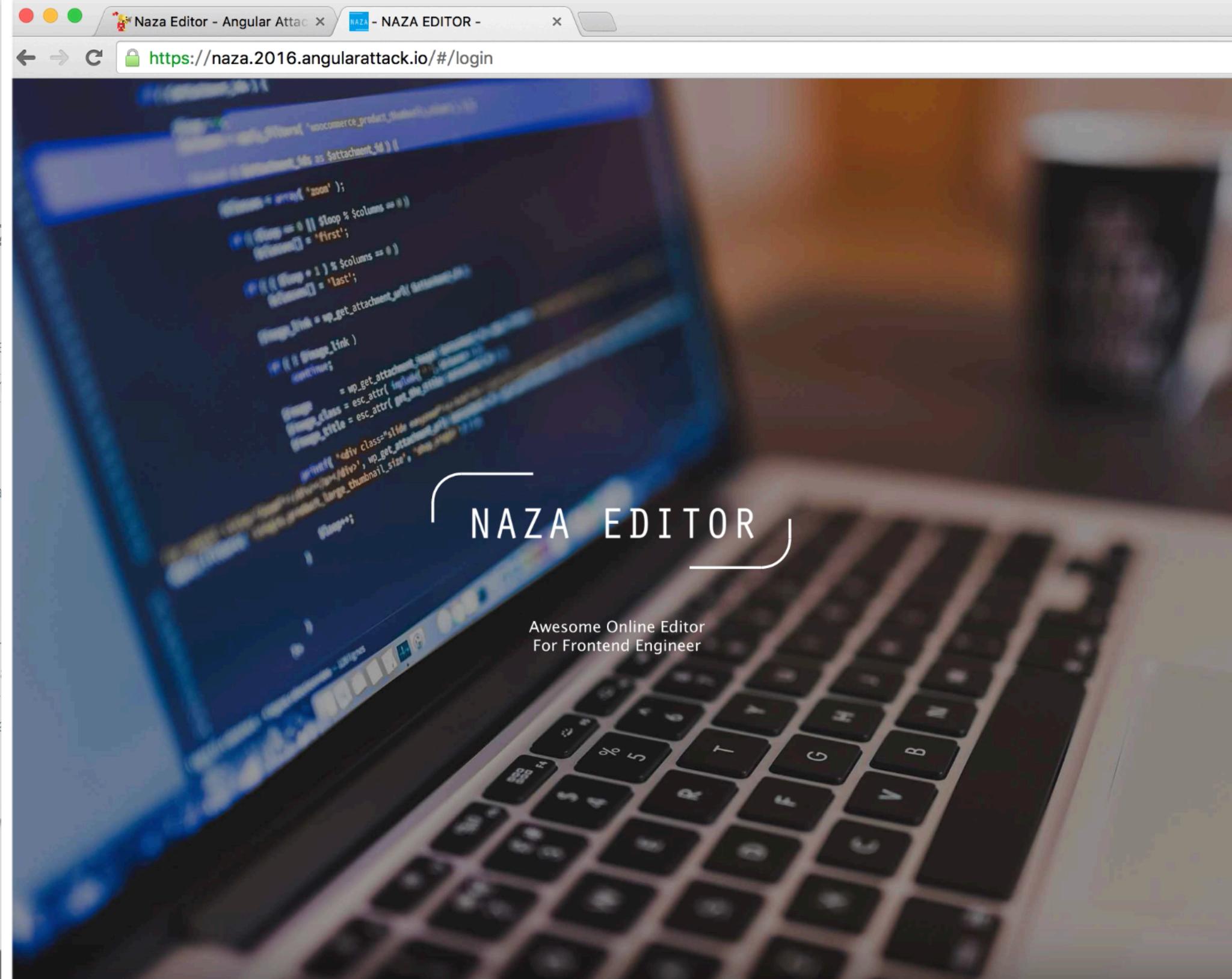
NAZA Team from China

**Team Members**

-  VTHINKXIE  
China
-  广彬-梁  
San Francisco
-  baqian  
Japan

**ENTRY URL**  
<https://naza.2016.angularattack.io/#/login>

**Naza Editor**  
Code online now, online editor



**Description**  
Naza Editor is a free platform for front-end engineers to code online.

**Instructions**

**Usage**

1. Sign Up and Login
2. Then you can create your project
3. You can edit your code
4. You can even create your own component

**Important**

**2016 IMPORTANT DATES**

REGISTRATION OPENS	MAR 1
COMPETITION	MAY 14-15
JUDGING	MAY 16-29
WINNERS	MAY 30

**PARTNERS**

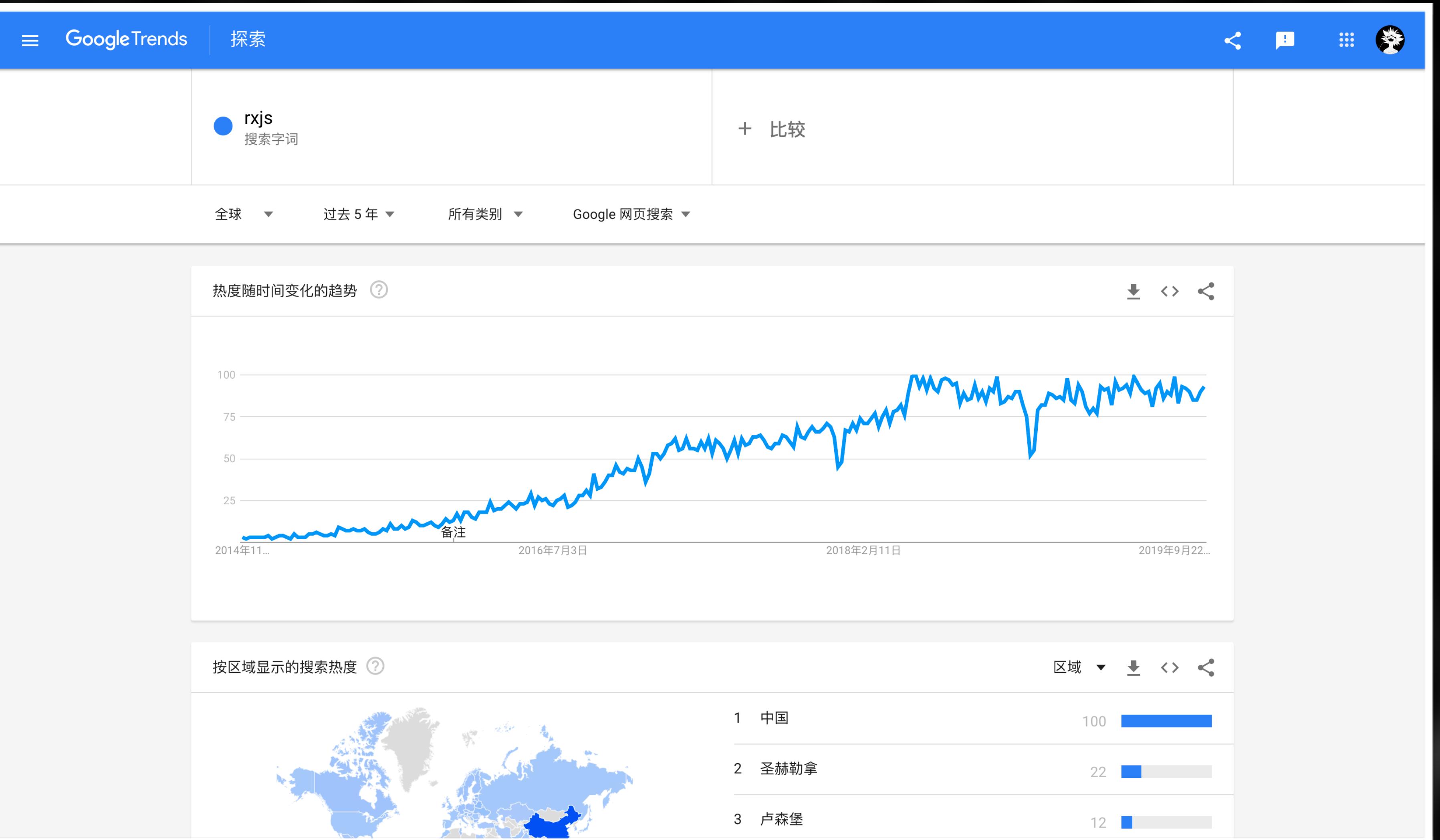
**Sign Up Success, Please login**

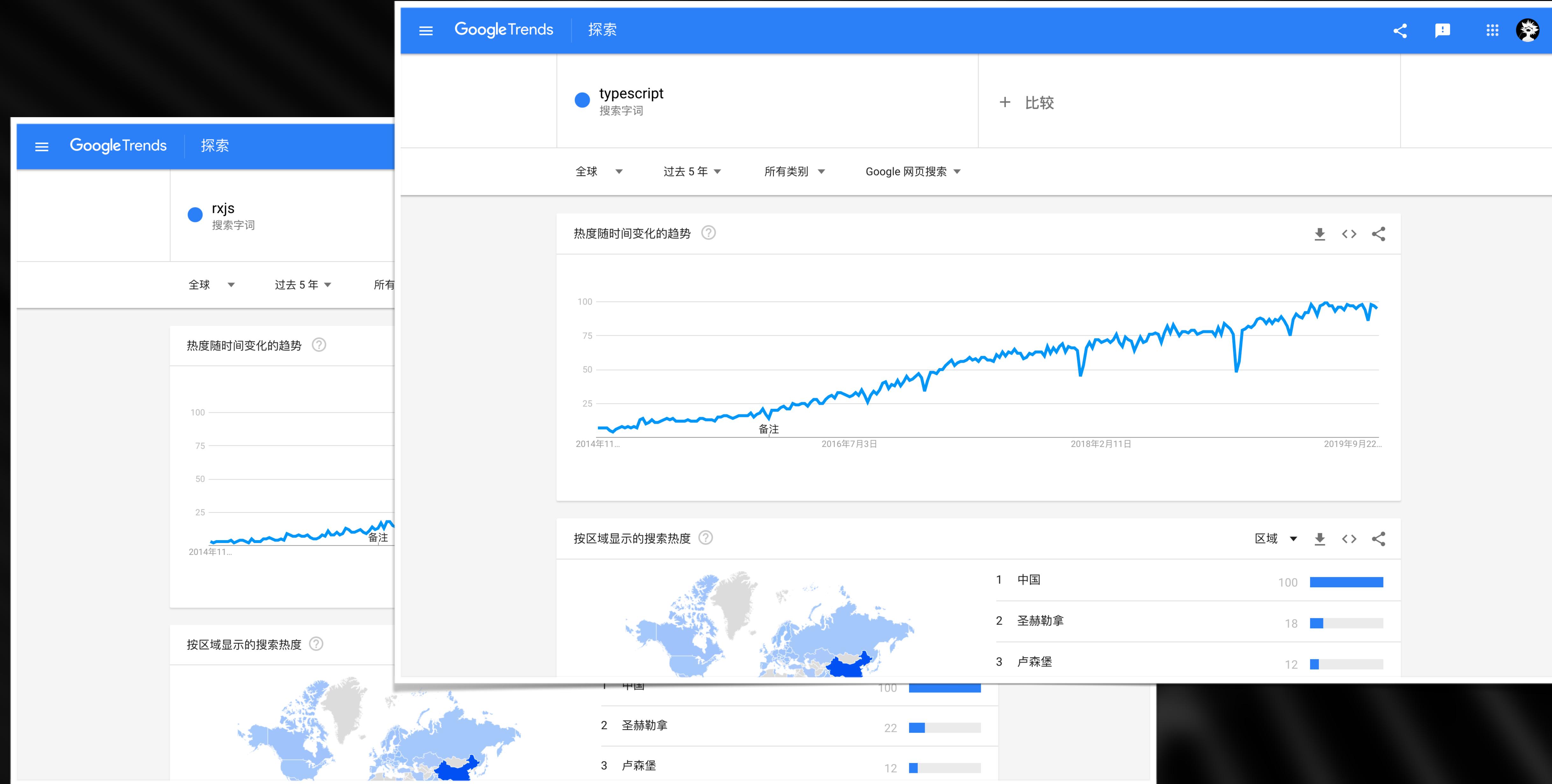
vthinktest  
vthinktest  
...  
...

**SIGN UP**

Have account already? Please go to [Login](#)







NG-ZORRO 自助服务群...

2099人



扫一扫群二维码，立刻加入该群。

# NG-ZORRO

# 钉钉答疑录

前端@杭州 简历发送

yadong.xyd@alibaba-inc.com